

IN THE CLAIMS

Please amend claims 43, 50, 52, 54, and 56, as indicated below, please cancel claims 44, 45, 51, 53, 55, 57, 58, 59, 63, and 101-108, and add new claims 109-128.

1-42. (Cancelled)

43. (Currently Amended) A system comprising:

a central processing unit (CPU) core;
a register file associated with the CPU core; a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator marks variables associated with the stack-based instructions in the register file as modified when the variables are updated as a result of the processing of the stack-based instructions to enable selective writing of the variables marked as modified to a memory; and

a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core. ~~with associated register file; and~~

~~a hardware accelerator operably connected to the central processing unit, the hardware accelerator adapted to convert stack-based instructions into register-based instructions native to the central processing unit, where the hardware accelerator marks the variables in the native CPU register file as modified when updated by the execution of Java byte codes.~~

44-49. (Cancelled)

50. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;
a register file associated with the CPU core;
a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator generates a new Java program counter due to a "GOTO" or "GOTO_W" byte code by sign extending the immediate branch offset following the "GOTO"

or “GOTO_W” byte code and adds it to the Java program counter of the current byte code instruction; and ~~with associated register file; and~~

a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core. ~~a hardware accelerator operably connected to the central processing unit, the hardware accelerator adapted to convert stack-based instructions into register-based instructions native to the central processing unit, where the hardware accelerator generates a new Java PC due to a “GOTO” or “GOTO_W” byte code.~~

51. (Cancelled)

52. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;

a register file associated with the CPU core;

a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator generates a new Java program counter due to a “JSR” or “JSR_W” byte code by sign extending the immediate branch offset following the “JSR” or “JSR_W” byte code and adding it to the Java PC of the current byte code instruction, computes the return Java program counter and pushes the return Java program counter onto an operand stack; and

a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core. ~~with associated register file; and~~

~~a hardware accelerator operably connected to the central processing unit, the hardware accelerator adapted to convert stack-based instructions into register-based instructions native to the central processing unit, where the hardware accelerator generates a new Java PC due to a “JSR” or “JSR_W” byte code, computes the return Java PC and pushes the return Java PC on to the operand stack.~~

53. (Cancelled)

54. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;
a register file associated with the CPU core; a hardware accelerator to process stack-based instructions in cooperation with the CPU core; wherein the hardware accelerator performs sign extension for the Java SiPush and BiPush byte codes and appends the sign extended data to the immediate field of a register-based instruction being composed based the stack-based instructions; and

a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core. ~~with associated register file; and~~

~~a hardware accelerator operably connected to the central processing unit, the hardware accelerator adapted to convert stack-based instructions into register-based instructions native to the central processing unit, where the hardware accelerator sign extends the SiPush and Bipush byte codes and appends to the immediate filed of the native instruction being composed.~~

55. (Cancelled)

56. (Currently Amended) A system, comprising:

a central processing unit (CPU) core;
a register file associated with the CPU core; a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator performs sign extension for the Java SiPush and BiPush byte codes and makes the sign extended data available to be read by the CPU core; and

a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core. ~~with associated register file; and a hardware accelerator operably connected to the central processing unit, the hardware accelerator adapted to convert stack-based instructions into~~

~~register-based instructions native to the central processing unit, where the hardware accelerator sign extends the SiPush and Bipush byte codes and made available to be read by the native CPU.~~

57-108. (Cancelled)

109. (New) The system of claim 43, wherein the hardware accelerator and the CPU core are within a CPU.

110. (New) The system of claim 43, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

111. (New) The system of claim 50, wherein the hardware accelerator and the CPU core are within a CPU.

112. (New) The system of claim 50, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

113. (New) The system of claim 52, wherein the hardware accelerator and the CPU core are within a CPU.

114. (New) The system of claim 52, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

115. (New) The system of claim 54, wherein the hardware accelerator and the CPU core are within a CPU.

116. (New) The system of claim 54, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

117. (New) The system of claim 56, wherein the hardware accelerator and the CPU core are within a CPU.

118. (New) The system of claim 56, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

119. (New) A system, comprising:

- a central processing unit (CPU) core;
- a register file associated with the CPU core; a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator maintains an operand stack for the stack-based instructions in the register file such that the operand stack in the register file define a ring buffer in conjunction with an overflow/underflow mechanism for moving operands in the operand stack between the register file and memory, and loads variables required for processing the stack-based instructions into the register file; and

- a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core.

120. (New) The system of claim 119, wherein the hardware accelerator and the CPU core are within a CPU.

121. (New) The system of claim 119, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

122. (New) A system, comprising:

- a central processing unit (CPU) core;
- a register file associated with the CPU core; a hardware accelerator to process stack-based instructions in cooperation with the CPU core; wherein the hardware accelerator maintains

operands and variables required for processing the stack-based instructions in the register file;
and

a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core.

123. (New) The system of claim 122, wherein the stack-based instructions comprise Java instructions.

124. (New) The system of claim 122, wherein the hardware accelerator and the CPU core are within a CPU.

125. (New) The system of claim 122, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.

126. (New) A system, comprising:

a central processing unit (CPU);

a register file associated with the CPU core; a hardware accelerator to process stack-based instructions in cooperation with the CPU core, wherein the hardware accelerator:

maintains an operand stack for the stack-based instructions in the register file such that the operand stack in the register file define a ring buffer in conjunction with an overflow/underflow mechanism for moving operands in the operand stack between the register file and memory, and loads variables required for processing the stack-based instructions into the register file,

generates a new Java program counter due to a "GOTO" or "GOTO_W" byte code by sign extending the immediate branch offset following the "GOTO" or "GOTO_W" byte code and adds it to the Java program counter of the current byte code instruction,

generates a new Java program counter due to a “JSR” or “JSR_W” byte code by sign extending the immediate branch offset following the “JSR” or “JSR_W” byte code and adding it to the Java PC of the current byte code instruction, computes the return Java program counter and pushes the return Java program counter onto the operand stack ,

performs a sign extension for the Java SiPush and BiPush byte codes and appends the sign extended data to the immediate field of a register-based instruction being composed based the stack-based instructions,

performs sign extension for the Java SiPush and BiPush byte codes and makes the sign extended data available to be read by the CPU core, and

marks variables associated with the stack-based instructions in the register file as modified when the variables are updated as a result of the processing of the stack-based instructions to enable selective writing of the variables marked as modified to a memory; and

a multiplexer to selectively connect the hardware accelerator and the memory to the CPU core.

127. (New) The system of claim 126, wherein the hardware accelerator and the CPU core are within a CPU.

128. (New) The system of claim 126, wherein the hardware accelerator processes the stack-based instructions in cooperation with the CPU core by converting the stack-based instructions into register-based instructions for execution in the CPU core.